SDxCentral SDN Controller Report

Brocade
Legal Disclosure

PLEASE SEE RISK FACTORS ON FORM 10-Q AND FORM 10-K FILED WITH THE SEC

This presentation, with prepared remarks, includes forward-looking statements regarding Brocade’s financial results, goals, plans, assumptions, strategy and business outlook, revenue, tax rate, cash, stock buybacks, OEM inventory, prospects, as well as U.S. Federal spending and order flow, and IT spending, which are only predictions and involve risks and uncertainties such that actual results may vary significantly. These and other risks are set forth in more detail in our Form 10-K for the fiscal year ended October 26, 2013 and our Form 10-Q for the fiscal quarter ended August 2, 2014. These forward-looking statements reflect beliefs, assumptions, outlook, estimates, and predictions as of today, and Brocade expressly assumes no obligation to update any such forward-looking statements whether as the result of new developments or otherwise.

In addition, this presentation includes various third-party estimates regarding market share and other measures, which do not necessarily reflect the views of Brocade. Further, Brocade does not guarantee the accuracy or reliability of any such information or forecast.

Certain financial information is presented on a non-GAAP basis. Management believes that non-GAAP financial measures used in this presentation allow stakeholders to gain a better understanding of Brocade’s comparative operating performance both from period to period, and to its competitors’ operating results. Management also believes these non-GAAP financial measures help indicate Brocade’s baseline performance before gains, losses or charges that are considered by management to be outside of ongoing operating results. Accordingly, management uses these non-GAAP financial measures for planning and forecasting of future periods and in making decisions regarding operations performance and the allocation of resources. The most directly comparable GAAP information and a reconciliation between the non-GAAP and GAAP figures are provided in the accompanying press release, which has been furnished to the SEC on Form 8-K and posted on Brocade’s website, and is included in the appendix in this presentation.
Market Landscape for Platforms

Gen1 Open Source

- Most are exclusively OpenFlow
- Academic experiments — no current market traction

“House” Controllers

- Strong market (Ops) interest/news coverage driven by corporate sales/market strength
- Relative product maturity
- Most focus on NW orchestration, not programmability
- Some tied explicitly to vendor’s own platforms

Gen2 Open Source

- Stated open source preference in many sectors
- Strong focus on programmability, interoperability
- Strong interest from developers & dev-heavy shops
- Some assembly currently required
OpenDaylight is an Open Source Software project under the Linux Foundation with the goal of furthering the adoption and innovation of Software Defined Networking (SDN) through the creation of a common industry supported platform.

**Code**
To create a robust, extensible, open source code base that covers the major common components required to build an SDN solution

**Acceptance**
To get broad industry acceptance amongst vendors and users:
- Using it directly or through vendor products
- Vendors using OpenDaylight in commercial products

**Community**
To have a thriving and growing technical community contributing to the code base, using the code in commercial products, and adding value above, below and around.
Defining “Open” in Open Source

As about Projects

• Who can contribute?
• Who does contribute?
• How are decisions made? Who can comment? Who can vote?
• What license does it use?

Ask about Products

• Does it integrate with other solutions from other vendors?
• Does it have an API?
• Does it follow open standards?
• Is it based on open source components?
• Does it upstream to open source projects?

“Simply stated, OpenDaylight is as open as open gets.”

http://www.opendaylight.org/blogs/2014/03/degrees-open
http://www.networkcomputing.com/50-shades-of-open-sdn/a/d-id/1234771
OpenDaylight Releases

• **Hydrogen** (February 2014)
  – 13 projects, 1.3m lines of code

• **Helium** (October 2014)
  – Clustering, AAA
  – DLUX UI
  – Policy-related projects
  – Karaf containers to customize packages

• **Lithium** (June 2015)
  – Improved OpenStack Neutron integration
  – Internet of Things
  – Service Function Chaining/NFV
  – Improved security, release processes
Beryllium Release Planning

• Focuses on
  – S3P (Security, Stability, Scalability and Performance)
  – Migration
  – HA/clustering

• Balances maturity (the above) with feature velocity
  – Some projects will be mature
  – Some of the Karaf features in mature projects will be stable
  – Stable features meet S3P, Migration, and HA/clustering requirements
  – Stable and “normal” distribution; stable only has stable features

• Emphasis on documentation, user stories, other support materials


© 2015 BROCADE COMMUNICATIONS SYSTEMS, INC.
Analysts See Momentum

“OpenDaylight is quickly evolving into something formidable with good potential for mainstream relevancy.”  
– Andrew Lerner, Gartner

“OpenDaylight may be the third center of gravity”  
– Andrew Lerner, Gartner

“OpenDaylight has become the Linux of network stacks: the foundation upon which both network vendors and users build the next generation of products and services.”  
– Kurt Marko, Forbes

An open source approach to software-defined networking (SDN) moved several steps closer this week to becoming a de facto standard.  
– Mike Vizard, IT Business Edge
OpenDaylight Users and Advisory Board

Deployed Users

- Telstra
- Tencent
- AT&T
- CERN Large Hadron Collider
OpenDaylight as the Common Platform

Companies that have announced products based on OpenDaylight

- ADVA
- BROCADE
- Ciena
- CISCO
- ERICSSON
- Extreme Networks
- IBM
- INOXYBE TECHNOLOGIES
- ORACLE
- CONTEXTREAM
One way to build a release based on OpenDaylight

- Clone the OpenDaylight code (it’s public, after all!)
- Modify the code to customize / brand
  - Add new code into the existing projects for your proprietary logic
- Run the build
- Test, Ship and Sell it!

- Great, that was easy…
- *This is the first thing nearly everyone does*
  - In practice, this results in a permanent fork of the codebase and eventually dropping out of the community
Treat OpenDaylight as a Third Party

• Don’t Download the OpenDaylight Source Code!
  – Treat OpenDaylight as a collection of read-only third-party artifacts
  – Reference ODL artifacts, don’t build them
  – Use Maven to access, maintain and manage your third-party dependencies for you

• For example:
  – We don’t recompile the basic java.util.ArrayList Java class; we reference/use it
  – We certainly don’t modify the java.util.ArrayList source code locally and build it
Leverage OpenDaylight Modularity

Isolate Proprietary Code

• Keep your proprietary code isolated to your own repositories, bundles, and Karaf features

• Leverage OSGi/Karaf/Maven to combine your code with OpenDaylight

• Leverage YANG/MD-SAL/Config system modularity to load proprietary implementations into the modeling system

• OSGi is generally friendly with the EPL license*

*Note: we are not lawyers. Be sure to discuss any legal / licensing issues with your legal team
The Result

• A product which references OpenDaylight artifacts

• Proprietary code is isolated, and can be updated without affecting OpenDaylight code
  – Also potentially avoid legal/licensing pain (ASK YOUR LAWYERS!)

• Changing versions of OpenDaylight code is just an update to a version in a pom file
  – …and some testing

• Allows for a stable OpenDaylight controller on which to build proprietary implementations
Build the Foundation and Frame Upstream
• Continuous build from ODL
• Contribute enhancements
• No forking or proprietary extensions
• Channel user requests into ODL workstream

Collaborative Innovation with Users
• Joint and custom development
• App design review, certification, and/or integration
• Developer tools and repo access
• Developer community resources
  – GitHub and Brocade community

Getting Started with SDN
• Architectural planning
• Stability and support for the whole controller environment
  – Support model assumes mixed-vendor environment
  – Full documentation
• Specialized education for operations and developers
Thank you (Red)
Where We Have Been… Where Are We Going?

The Control Plane Evolution

- PSTN: Single Service – Single Control Plane
  - ca. 1960

- Multimedia Communications and Services
  - Multiple Services – Multiple Control Planes
  - 1980’s – 2010’ish

- Ubiquitous Multimedia Communications and Services
  - Multiple Services – Single Control Plane
  - Today – Tomorrow

Software Defined Network
Modern, Inclusive SDN

- Device-by-device operation
- Proprietary, vendor-specific vertical stacks for control, management and orchestration
- Limited innovation in individual silos

- Network-wide operation
- Open control, management and orchestration using open control protocols/modeling langs
- Independent innovation at each layer of the stack
Why Open Source?

• Avoid vendor lock-in

• Have a seat at the table

• Faster Innovation

• Interop & Integration